Integración de un periscopio militar a un simulador de realidad virtual para entrenamiento táctico

José A. Marone[1], Marcelo A. Tosini[1]

[1] Grupo de Sistemas Digitales, Instituto INTIA, Facultad de Cs Exactas, UNCPBA, Tandil, Buenos Aries, Argentina
{marone, mtosini}@exa.unicen.edu.ar

Resumen. El presente trabajo describe tareas de actualización recientemente realizadas en la Escuela de Submarinos y Buceo (ESyB) de la Armada Argentina con sede en la Base Naval de Mar del Plata en el adiestrador de operarios de submarinos en el marco del proyecto de transferencia tecnológica (RCS 1112/13 UNCPBA-ESyB). En particular se trabajó en la modernización de un simulador de periscopio, desarrollado originalmente por este mismo grupo de investigadores en el año 2003. Los distintos movimientos y controles asociados a los comandos del periscopio son convertidos a una interfaz clara de alto nivel y enviados al motor de simulación desacoplando así ambos sistemas. Este artículo muestra un panorama general de la arquitectura del sistema de control del periscopio y describe las soluciones implementadas a nivel de hardware y software. Se consiguió implementar un sistema embebido adecuado al presupuesto, con componentes accesibles a nivel nacional y con tiempos de respuesta por debajo de lo exigido por el motor de simulación.

Palabras clave: Sistemas Embebidos, Actualización de Sistemas Militares, Codiseño de hardware/software

1 Introducción

Sin dudas la actualización o incorporación de nuevas tecnologías en sistemas antiguos no es una tarea simple. En muchos casos, y particularmente en sistemas de defensa, se debe interactuar con sistemas de diversas índoles (eléctricos, electrónicos, mecánicos, electromecánicos, ópticos, etc.) haciendo que las tareas de actualización y mantenimiento sean complejas y costosas [1]. En muchos casos los países en vías de desarrollo no pueden afrontar la compra o el desarrollo de equipos de entrenamiento nuevos, por lo que deben modernizar gradualmente los más antiguos [2].

En las últimas décadas se ha producido un vertiginoso crecimiento de los simuladores de realidad virtual para entrenamiento de operarios, en parte, debido al incremento en la potencia de cálculo de las computadoras modernas y sus tarjetas gráficas. Para que el entrenamiento sea efectivo los operarios deben disponer de la funcionalidad equivalente y en lo posible sentirse en el ambiente real, para lo cual se debe recurrir a la simulación de los procesos físicos intervinientes. En el caso de los simuladores de equipos militares es deseable que las interfaces de comando se mantengan tal cual se encuentran en el momento de operación, por lo que se utilizan

los comandos reales que deben ser adaptados por medio de sistemas electrónicos al motor de simulación [3].

En un sentido amplio, "Sistema embebido" es la denominación aplicable a los equipos electrónicos que incluyen procesamiento de datos, pero que, a diferencia de una computadora de propósito general, están diseñados para satisfacer una función específica, como en el caso de un reloj digital, un router, el sistema de control de un automóvil (ECU), un reproductor de mp3, etc. Componentes fundamentales de un sistema embebido son la arquitectura de hardware subyacente, los dispositivos de entrada-salida y el software de control que en general esta optimizado para funcionar en sistemas de tiempos de respuesta acotados, bajo consumo y alta fiabilidad [4].

En este trabajo se presenta el desarrollo de una sistema embebido para la actualización y adaptación de un periscopio militar, y su vinculación al motor de simulación de realidad virtual, desarrollado para el entrenamiento de los alumnos de la Escuela de Submarinos y Buceo de la Armada en la ciudad de mar del plata [5], los requerimientos fueron analizados cuidadosamente, ya que se requería realizar tanto adaptaciones electrónicas como eléctricas y mecánicas, manteniendo el costo de actualización bajo y con suministro de repuestos accesibles en el país.

Por otro lado era prioritario no modificar la dinámica de uso ni la apariencia del equipo ya que el periscopio debe seguir luciendo y funcionando tal como los originales instalados en los submarinos.

2 Descripción de Sistema

La operación de un submarino involucra el manejo de numerosos subsistemas de control tales como el de profundidad, de timón y planos, de potencia de motores y de lastre, por otro lado están los subsistemas tácticos, que incluyen sonares, computadoras de tiro y los periscopios. Este último sistema es usado en tareas de vigilancia, detección de buques y ataque. En la ESyB se desarrolló un simulador de este sistema táctico, llamado ADITACSUB, que en su gran mayoría utiliza el mismo equipamiento que se encuentra en una nave real, (periscopio de ataque, sonar, computadora de cálculo de tiro, mesa de táctica, etc.), como se aprecia en la Figura 1.

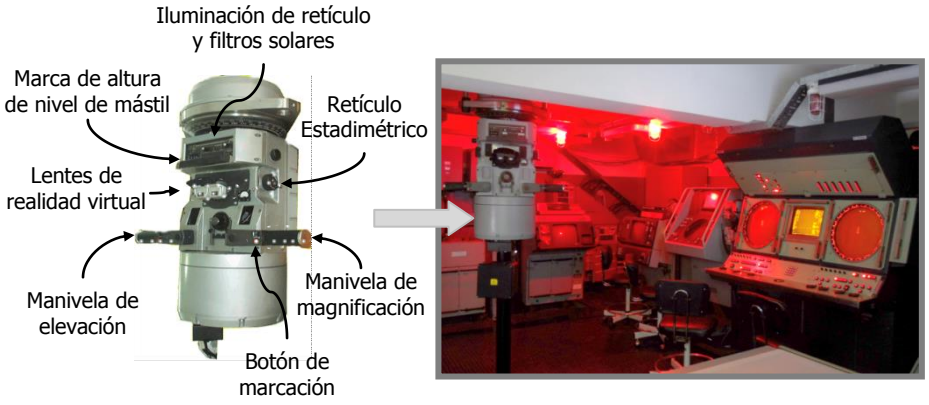

Figura 1: Descripción de las partes del periscopio y sala de simulación

Las señales que recibe cada uno de estos equipos son simuladas con computadoras, lográndose un grado de realismo aceptable. Estas computadoras, en la sala de generación de escenarios, permiten generar blancos tanto navales como aéreos de diferentes tipos y con parámetros de posición, orientación y velocidad determinados; y generar los parámetros dinámicos del propio submarino (posición, rumbo, profundidad, velocidad, etc). En uno de los procesos de creación de este simulador se adaptó un periscopio real de submarino (a la izquierda de la figura 1 y en el lado izquierdo de la sala de simulación) [5], en el que se reemplazó el visor y las ópticas por unos lentes de realidad virtual sobre el cual se proyecta un modelo de visualización tridimensional generado en tiempo real mediante un simulador gráfico de alto desempeño. De este modo, las operaciones y movimientos ejecutados sobre el periscopio son transformados en señales digitales e ingresados al simulador para realimentar el modelo visual. La Figura 2 muestra el esquema general del simulador y el modelo de comunicación.

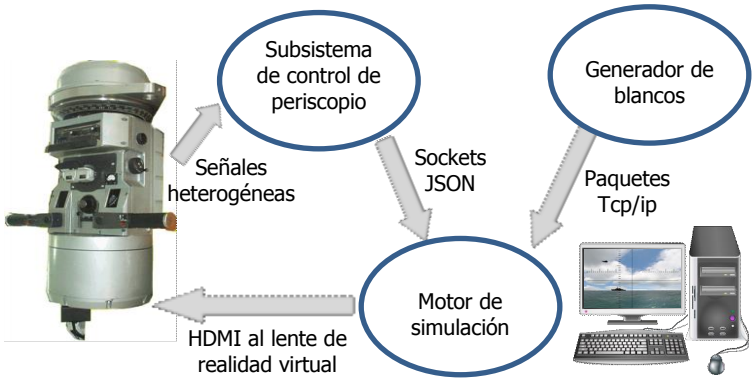

Figura 2: Modelo del simulador ADITAC

Este trabajo se centra en el "subsistema de control de periscopio" este módulo debe encargarse de adecuar la señales eléctricas como así también los movimientos mecánicos provenientes del periscopio y presentarlos de una manera clara ante el motor de simulación. Las operaciones más importantes que deben censarse son: Giro en azimut, selección de ópticas, control de elevación, filtros para disminuir intensidad de luz recibida, control de iluminación del retículo, desfasaje de retículo estadimétrico; y se describen a continuación en la tabla 1.

Tabla 1. Descripción de las operaciones del periscopio implementadas en el simulador ADITAC. Cada función requirió un análisis de rangos de uso y precisión mínima necesaria.

Operación	Descripción
Rotación en Azimut	Permite el giro del periscopio en 360º sobre su eje vertical sin límite de vueltas en ambos sentidos. La posición de 0º es tal que el periscopio mira hacia la proa del buque.
Manivela de elevación	Elevación de la óptica de visualización en el rango -10º a 60º de modo de realizar un paneo vertical del área de visión marítima, terrestre o aérea.

Cambios de aumento	Indicador de tres grados de aumento -½X, 6X y 12X.
Retículo estadimétrico	Desdobla la imagen con el objeto de estimar distancias a partir del uso de trigonometría, desplazando una imagen de un buque sobre si misma de manera que la parte inferior del buque coincida con el extremo superior del mástil mayor; y conociendo la altura de palo de dicho buque.
Panel de altura	Permite introducir al sistema la altura de mástil del objetivo. Se obtienen tres dígitos BCD desde un panel numérico mecánico.
Filtros solares	Cuatro grados de oscurecimiento del sistema óptico.
Rebatido de manivelas	Indicador de posición de las dos manivelas laterales. La posición hacia debajo de las dos manivelas indica a los sistemas que el periscopio está operativo.
Iluminación de retículo	Cinco grados de iluminación de la retícula de medición en pantalla.
Botón de marcación	Permite al operario marcar un blanco informando su posición al sistema de sonar.

3 Diseño e implementación de la solución

A fin de generar un modelo de solución funcional y correcto se utilizó la metodología de co-diseño en "V" para sistemas embebidos en la cual la verificación se desglosa en etapas de nivel de abstracción creciente. En cada etapa de diseño se crea un plan de pruebas que es el que guía la etapa de validación que le corresponde, este trabajo no intenta seguir la metodología como un formalismo sino más bien como una guía para el desarrollo ordenado.

3.1 Diseño de Software

El diseño del software tiene tres partes bien diferenciadas:

1. Diseño del software del colector de datos (Sistema embebido)
2. Diseño del software del controlador del periscopio (Driver)
3. Diseño del protocolo de envío y recepción de datos

La Figura 3 esquematiza la interacción entre el software del *Colector de Datos* y del *Controlador de Periscopio* como así también la conexión con el motor de simulación, en el diagrama se observa también los protocolos e interfaces de comunicación con el resto del entorno.

Diseño del software del controlador

Para el diseño del sistema de control de, se utilizó una arquitectura de software centrada en datos tipo blackboard [7], proponiendo dos threads principales, uno de ellos actualiza y transforma las señales de datos provenientes de las interfaces electrónicas del periscopio, el otro informa a las rutinas correspondientes sobre la actualización de un determinado conjunto de datos. Cada una de estas rutinas

empaqueta los datos y los encola para ser enviados por el canal serie al controlador de periscopio, para evitar inconsistencias en los datos se desarrollaron los mecanismos de sincronización correspondientes.

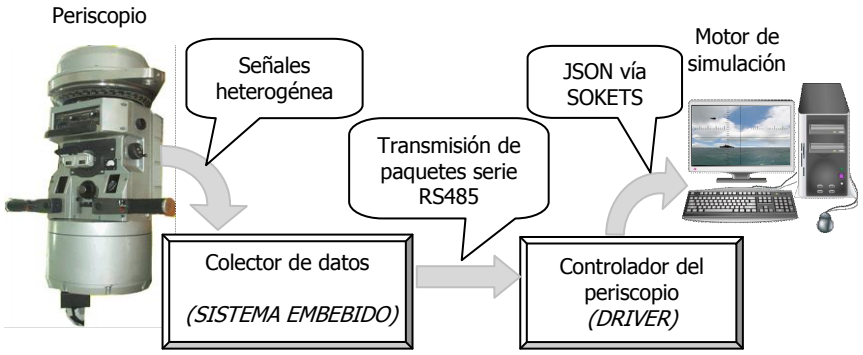

Figura 3: Esquema de interacción de módulos de software

Los datos son mantenidos en la memoria principal y actualizados en un período menor a 1ms, cabe destacar que todas las tareas de procesamiento de datos son determinísticas y sin bloqueos.

Los datos son considerados válidos para ser transmitidos si se cumplen las siguientes reglas:

1. El dato fue actualizado desde la última vez que fue transmitido o en caso de inicio del sistema si estaba como *no_inicializado*.
2. El dato no está siendo actualizado en ese momento.

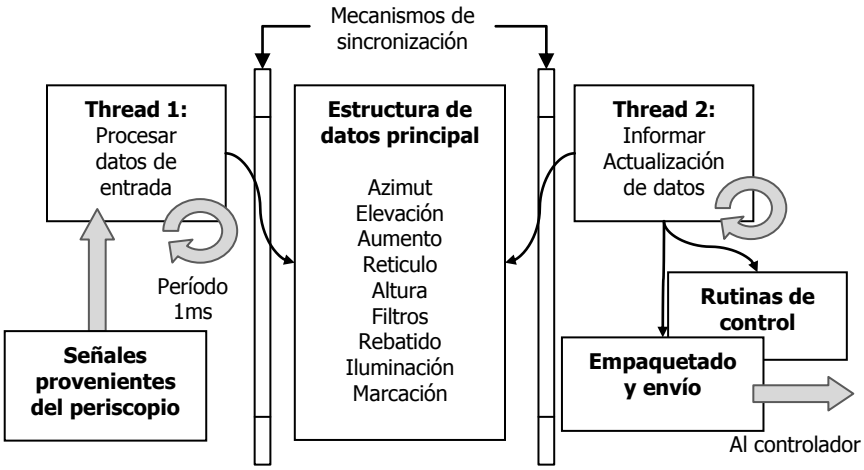

Figura 4: Esquema general de los módulos de software embebido.

Diseño del software de control del periscopio

El software de control (comúnmente denominado driver) es un pequeño sistema (ver **Figura 5**) que se inicia al encender el motor de simulación y permite realizar las siguientes funciones:

1. Chequeo del estado de salud del colector de datos y la comunicación
2. Configuración de los parámetros iniciales
3. Visualización de los datos en formato legible
4. Monitorización del sistema de recepción y despacho de eventos

El diseño es una típica arquitectura MVC (Model View Controller) que intercambia mensajes con el colector de datos vía una comunicación serial.

Para definir una interfaz clara con el equipo que desarrolló el motor de simulación se optó por una arquitectura *Cliente-Servidor* basada en *sockets.* El formato elegido para los datos fue JSON (JavaScript Object Notation), quedando de esta manera totalmente desacoplados ambos desarrollos. Esto impactaría positivamente también en el testeo e implantación del sistema.

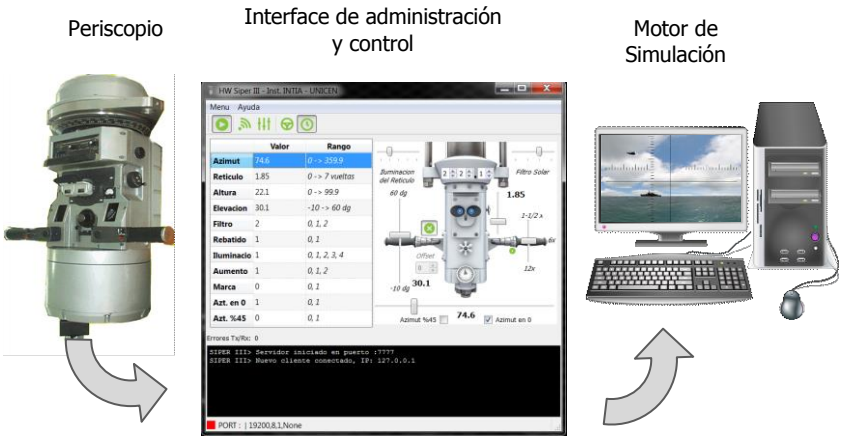

Figura 5: Interfaz gráfica del controlador de Periscopio SIPERIII

Diseño del protocolo de envío y recepción de datos

Luego de analizar varios métodos de transmisión de información, se optó por la implementación de uno propio sobre la capa física que brinda el standard RS485 [8]. El protocolo se basa en MODBUS RTU [9] al que se le han recortado varios campos para acelerar la tasa de transmisión, la definición del paquete de comunicación es compartida tanto por el colector de datos como por el controlador de periscopio.

Cuando ocurre un nuevo evento en los mandos del periscopio, el colector de datos lo decodifica e inmediatamente envía un paquete con dicha información al controlador de periscopio, este verifica su correctitud y procede a desempaquetarlo para encolarlo en la cola de eventos que se enviarán al motor de simulación.

Cabe destacar que cada paquete viene acompañado de un campo de chequeo de correctitud por códigos de redundancia cíclica (CRC). En la **Figura 6** se presenta el paquete de datos típico de comunicación entre el colector de datos y el controlador de periscopio.

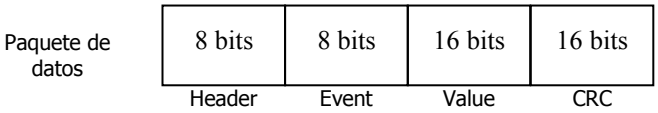

Figura 6: Estructura del paquete de datos básico.

3.2 DISEÑO DE HARDWARE

El diseño del hardware del sistema se dividió en dos etapas. Por un lado, se realizó el análisis de requerimientos y las adaptaciones necesarias a fin de que interactúen con los sensores y actuadores seleccionados. En esta primera etapa, las adaptaciones realizadas para las operaciones fueron las siguientes:

Rotación en azimut: El sistema de giro en 360º del periscopio posee una caja de engranajes en la parte superior que interactúan con varios sincromotores. En esa caja se realizó la Inserción mecánica de los sensores. Se utilizó un encoder relativo de 360º con una sensibilidad de 60 pulsos por grado (error de un minuto de grado) y dos sensores de paso por 0º y por múltiplos de 45º. Estos últimos sensores permiten detectar posiciones absolutas del periscopio.

Manivela de elevación: Para esta función se realizó el reemplazo del sincromotor ubicado dentro de la manivela izquierda por un potenciómetro con mapeo directo de la posición angular de 70 grados a un valor binario de 10 bits vía un conversor analógico-digital. Se utilizó un potenciómetro lineal de altas prestaciones para disminuir el posible ruido eléctrico en la conversión. Los requerimientos originales planteaban errores máximos de medio grado. Mantener bajo el ruido de digitalización en este componente es importante cuando se usa el aumento de 6x ó 12x

Cambios de aumento: En este caso se reutilizó el componente original consistente en un selector rotativo de tres posiciones con tres salidas digitales de 0 ó 5 voltios asociadas a cada uno de los tres aumentos.

Retículo estadimétrico: Esta función opera mediante una rueda lateral con capacidad de giro de siete vueltas. En este caso se realizó la unión mecánica entre la cremallera interna asociada a la rueda de control mediante una unión cardánica a un potenciómetro multivueltas de altas prestaciones y su posterior digitalización mediante un ADC de 10 bits. La exactitud debida a errores inducidos por ruido eléctrico es de 8 bits.

Panel de altura*:* Se reutilizo el panel original consistente de tres selectores mecánicos de dígitos decimales. Puesto que la salida de cada selector es un valor BCD de 4 bits, se envían los tres valores directamente a la placa de lectura de datos.

Filtros solares: Se adaptó en este caso un selector rotativo de cuatro posiciones que devuelve los valores codificados en binario en dos bits.

***Rebatido de manivelas**:* El rebatido de las manivelas laterales del periscopio aporta información al resto de los sistemas tácticos indicándoles que deben recibir datos de posición y tipo de los blancos detectados. La adaptación en este caso consistió en la detección de la posición individual de cada manivela en sus dos posiciones extremas (rebatida o extendida). Para la detección se utilizaron sendos conmutadores eléctricos de presión conectados en serie.

Iluminación de retículo: Para esta funcionalidad se adaptó un selector rotativo de cinco posiciones al cuerpo del periscopio que devuelve las intensidades seleccionadas codificadas en binario en tres bits.

***Botón de marcación**:* Se utilizó el componente original consistente de un pulsador que devuelve 5 voltios en estado de reposo y 0 voltios al ser oprimido (lógica negativa).

***Encendido/apagado de los lentes de visualización y sistema de ventilación**:* Esta funcionalidad no existe en el periscopio real puesto que tiene un sistema de visualización óptico. Se implementa en el simulador a fin de encender los distintos subsistemas cuando se inicia una sesión de entrenamiento.

Una segunda etapa del diseño de hardware consistió en la selección del controlador apropiado para el manejo de las señales descritas anteriormente y su procesamiento y empaquetado para transmitirlas al driver en la PC. Se optó por un sistema de desarrollo basado en un microcontrolador con arquitectura AVR [10] de la firma ATMEL más una placa de diseño propio para soporte de potencia y comunicación serial vía protocolo rs485 que aumenta la inmunidad al ruido eléctrico en un ambiente donde existen tensiones de alimentación alternas.

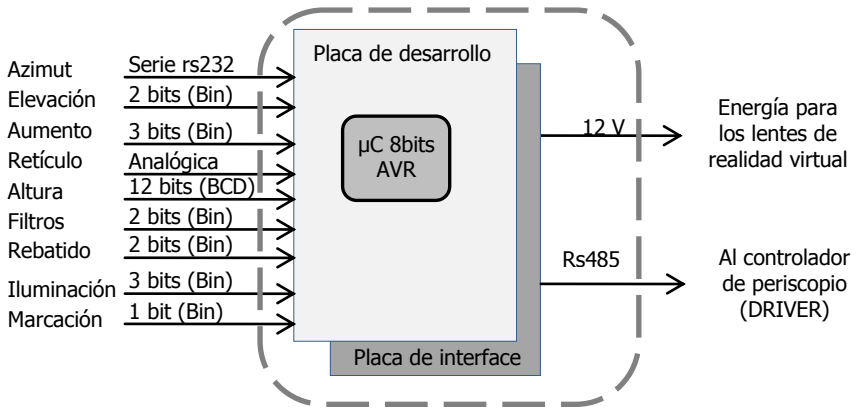

Figura 7: Esquemático del hardware del colector de datos

Los requerimientos originales de implementación del sistema con determinadas restricciones presupuestarias llevaron al uso de un encoder relativo para la adquisición del ángulo de azimut. Debido a este tipo de encoder, no puede conocerse la posición absoluta del periscopio. Una solución parcial es la incorporación de uno o más sensores de posición para detectar ángulos determinados (sensor de 0º y sensor de

múltiplos de 45º). Esta solución, aunque efectiva, es falible si se usa el periscopio durante cierto tiempo sin pasar por algún sensor absoluto. Para solucionar el problema se implementó un filtro de corrección que disminuye gradualmente el error a fin de evitar efectos de saltos en la visualización.

4 Experimentos

El subsistema de control de periscopio debe operar en un ambiente con dos desventajas importantes. Por un lado, la computadora donde se encuentra el controlador de periscopio y el sistema gráfico (en la sala de control de ejercicio) está ubicada a aproximadamente diez metros del colector de datos (dentro del periscopio en la sala de adiestramiento) y, los demás sistemas en esa sala funcionan con tensiones alternas altas. En este ambiente es necesario asegurar que el ruido eléctrico no perjudique la calidad de la comunicación. A fin de probar el desempeño de la etapa de comunicaciones, y dado que el periscopio real se encuentra ya instalado en la base naval, se desarrolló un emulador en hardware del mismo para la realización de pruebas de laboratorio sin necesidad de conectarse físicamente al sistema.

Los experimentos realizados permitieron determinar la cantidad de errores de transmisión de paquetes con distancias superiores a diez metros. En estas pruebas se determinó que los errores fueron mayormente de sincronización durante el encendido o apagado del periscopio y desaparecían cuando las máquinas de estado de transmisión y recepción se sincronizaban.

En pruebas de stress se sometió al sistema a sesiones de funcionamiento de hasta una semana completa para comprobar la eficiencia de las comunicaciones y los componentes de hardware con resultados satisfactorios.

En los ensayos de campo, en el ADITAC, se detectaron errores relacionados con ruidos de conversión en los ADC del colector de datos debidos a las perturbaciones eléctricas de los sistemas. Esto obligó al uso de un sistema de comunicación optoacoplado entre el colector de datos y el controlador del periscopio.

5 Conclusiones y trabajos futuros

Se consiguió implementar un sistema embebido adecuado al presupuesto, con componentes accesibles a nivel nacional y con tiempos de respuesta por debajo de lo exigido por el motor de simulación.

La adecuada división entre las etapas de hardware y software, sumado a un modelo de comunicación integral permitió un proceso de desarrollo casi en paralelo de ambas etapas y, al mismo tiempo, el protocolo de comunicación desarrollado puede ser ampliado para su aplicación en otras aplicaciones.

Terminada la experiencia de implantación de este sistema de entrenamiento táctico se comenzó con las primeras etapas para el desarrollo de un nuevo simulador de operaciones de manejo integral de submarinos para entrenamiento en tareas de gobierno y control de inmersión.

Este emprendimiento busca afianzar lazos de trabajo interdisciplinario e interinstitucional; y permitirá aplicar metodologías de diseño más robustas para afrontar proyectos de mayores dimensiones.

Agradecimientos

Este trabajo se desarrolló dentro del marco del proyecto de investigación "Metodologías de Diseño para Sistemas Embebidos" 03-C238 del programa de incentivos a la investigación y el PICT-2009-0041: Desarrollo y Verificación de Sistemas Digitales Complejos. Además, fue financiado en parte por el proyecto de transferencia tecnológica (RCS 1112/13 UNCPBA-ESyB).

Referencias

1. Loffler, T., Nielson, J.: International HARM precision navigation upgrade. A GPS/INS missile upgrade that improves effectiveness and minimizes friendly-fire accidents. IEEE Aerospace and Electronic Systems Magazine, 18(5):26-31, (2003).
2. Aguierre, L., Ramirez, D., Leiva, L., Marone, J., Vazquez, M.: AHRS R-001: Actualización de Sistemas Inerciales de Navegación en Aeronaves Supersónicas. II Congreso de Microelectrónica Aplicada. (2011).
3. BORONI G., VÉNERE, M. "Un simulador distribuido para entrenamiento de operarios". Proceedings VIII Congreso Argentino de Ciencias de la Computación. 2002. ISBN Nº: 987-96- 288-6-1. pp. 727-738
4. Tosini, M., Todorovich, E., Vázquez, M., Leiva, L., Aciti, C., Marone, J., Goñi, O., Pantaleone, L., Acosta, N., Curti, H., Toloza, J., Kornuta, C: Metodologías de Diseño para Sistemas Embebidos. XV Workshop de Investigadores en Ciencias de la Computación. (2013)
5. Boroni, G., Vagliati, P., Venere, M., Marone, J., Tosini, M., Avila, E., Grasso, O., Lagar, D., Leisamon, R.: Realidad Virtual y Simulación Computacional Aplicada al Entrenamiento de Operarios. Mecánica Computacional, Vol XXII, Bahia Blanca. Noviembre 2003
6. Pérez, A., Berreteaga, O., Ruiz de Olano, A., Urkidi, A., Perez, J.: Una metodología para el desarrollo de hardware y software embebidos en sistemas críticos de seguridad. Sistemas, Cibernética e informática. Vol 3, Nro 2, ISSN: 1690-8627. (2006).
7. Daniel D. Corkill. 1988. Design Alternatives for Parallel and Distributed Blackboard Systems. Technical Report. University of Massachusetts, Amherst, MA, USA.
8. Soltero, M., Zhang, J., Cockril, C., Zhang, K., Kinnaird, C., & Kugelstadt, T. RS-422 and RS-485 Standards Overview and System Configurations, Application Report, (2002).
9. MODBUS.ORG, "MODBUS Application Protocol Specification V1.1," www.modbus.org, Access Date, Jan, 2003.
10. Myklebust, G: The AVR Microcontroller and C Compiler Co-Design. ATMEL Development Center, Trondheim, Norway, (1996).